```
                    AP
        : implementation file


#it
#defi
#undef
static ch
#endif

////////////////                          //////////////////////////
// APostOffice

MPLEMENT_DYNCREATE(AP

    office::APostOffice()

        ableAutomation();

            keep the applicatio                              tomation
              object is active, th                           kApp.

                (NULL,"PostOffice",

A
{
                                  ation when all objec
                                     the destructor calls

//      ::M                              "PostOffice",MB_OK);

        AfxOleU
}

void APostOffice::OnFin
{
        // When the last re                              ject is released
        // OnFinalRelease is                               automatically
        // deletes the object.                             uired for your
        // object before calling

          MessageBox(NULL,"OnFinalRe.

              get::OnFinalRelease();


                  tOffice, CCmdTarget)
                    APostOffice)
                      the ClassWizard will add a                    e.

EN

BEGIN_                              mdTarget)
                                    fice)
        Di                              tName", GetName, VT_BSTR,
                                        ge 1
```


EXHIBIT A

```
                                        APo
                    TON(APostOffice, "GetLo                                    VTS_VARIANT

                        APostOffice, "Start", S
                          stOffice, "Stop", Stop
                            Office, "Create", Crea
                              fice, "GetComputer", G
                                ce, "Handshake", Hands
    VTS_
                                    "MapCErr", MapCErr, VT_
                                     apCOut", MapCOut, VT_BO
                                       se", Close, VT_BOOL, VT
            //}}
    END_DISPATCH_M

    // Note: we add su                                  support typesafe bindin
    //  from VBA.  This                                   is attached to the
    //  dispinterface in t

        C57D0B-CE64-11CF-8BA
        onst IID IID_IAPostO
            0b, 0xce64, 0x11cf,                                  0x1a, 0xe7, 0x8e } };

            E_MAP(APostOffice, C
            E_PART(APostOffice,
            )

            -8BAF-0000E81AE78E}
            tOffice, "Eliza.Postc                                        0x11cf,
              0x1a, 0xe7, 0x8e)

    //                  ''//////////////////////,
    //

    BOOL APo                          e, long Timeout)
    {
            retu                          Timeout);
    }

    BOOL APostOffice::
    {
            return CPostO
    }

    STR APostOffice::GetName()

            CString strResult=CPost
            return strResult.AllocSys


            fice::GetLocalIds(const VA

                ept only array declared as
                (VT_ARRAY|VT_BYREF|VT_I2),
                * psa=*Ids.pparray;

                    rayAccessData(psa,(void**)
                    GetLocalIds(pData,Max);
                    (psa);

    }


    BOOL APost
```

```
                                        APo
                        ffice::StartTraffic();

                                Timeout)
                                    Traffic(Timeout);
}

BSTR APostO
{
        // TODO                                  ode here
        char buf                                  ]="";
        DWORD leng=
        GetComputerNa
        CString strResu
        return strResult;

        ffice::Handshake(LPC                                  onst VARIANT FAR& Ids,

                ept only array decla                                  "
                (VT_ARRAY|VT_BYREF|V
                * psa=*Ids.pparray;

                    ayAccessData(psa,(vo                                  E;
                    ffice::Register(Name,
                        (psa);
}

BOOL APostO                             hort meth)
{
        return                              eth);
}

BOOL APostOffice::Ma                            )
{
        return CPostOffi
}
```

APo

header file

```
//////////////////////.

// A

class __                    ce : public CCmdTarget,
{
        DECL

// Attributes
public:

Operations
ic:

        APostOffice();                                  used by dynamic creation
          tual ~APostOffice(),

              izard generated virt
              TUAL(APostOffice)

                 FinalRelease();

pr

                          nctions

                          rd will add and remov
        //}}.

        DECLARE_
        DECLARE_OL

        // Generated O
        //{{AFX_DISPATCH
        afx_msg BSTR GetNa
        afx_msg short GetLoc                    , short Max);
        afx_msg BOOL Start();
        afx_msg BOOL Stop(long
        afx_msg BOOL Create(LPCTS
        fx_msg BSTR GetComputer();
            msg BOOL Handshake(LPCTST                   nst VARIANT FAR&
            );
              BOOL MapCErr(short rcvr,
              OL MapCOut(short rcvr, sh
                 Close(long Timeout);
              TCH
                 H_MAP()
                    MAP()

//.                       ////////////////////////,
```

```
                                        CPo

                                  "

class CM
{
        CPostO
        WORD rcv
        CString li

hlic:

        CMsgStream(CPostOf                                  eth)
                : po(_po), rc


                buf:
                  underflow() { retur
                  verflow(int c)

                      r(c);
                        {
                          nd(CMsg(0,rcvr,meth,L
                              ";


};

//==============                                  ================

CRcvr::CRcvr(WORD _i
        : id(_id), po(_
{
        cerr=::cerr;
        cout=::cout;

        CRcvr()

                ========================

                fice(int nLcl, int iLcl,
                  r>(nLcl,iLcl), clients(nR
                    L), cout(NULL)

                          nt(NULL,TRUE,TRUE,NULL);
}

CPostOf
{
        bEn
                                          e 1
```

```
                                        Ch
                ngleObject(hThreadEnded,
                  hThreadEnded);



b                           rcvr, WORD meth)
{
                                  this,rcvr,meth);
        I                           {


        }
        return TRUE;

    CPostOffice::MapCOut(

        elete cout;
        out=cout=new CMsgStr
            nt i=0; i<GetSize();
                CRcvr* pr=GetAt(i);
                f(pr->IsLocal())
                        pr->cout=cout;


b                           TR lpszName, DWORD tim
{
                                  me);
}

int CPostOffi
{
        for(int i
                if(                          ;
        return -1;
}

  CPostOffice::GetLocalIds(

        int cnt=0;
        for(int i=0; i<GetSize();
                if(GetAt(i)->IsLocal
                        *Ids++ = GetAt
                cnt;

                ndTo(int handle, CMsg& msg,

                    ) && (handle<GetSize()) &&                    );

//
//
//

class __de                          public CRcvr
                                    ge 2
```

```
                                        CP

                        -lient;


                                    parent, WORD id, CPostC
    CP
        {                                                   -q);
        }
        BOOL IsLoc
        void Idle()


        -office::Register(LPC_                                  -RD Ids, int Size)

            -eate new client
             -ient* pcli=new CPipeC
                 establish a link to
                 -nen(Computer,Name))
                    -n 0;
                        -i);
                            -ze; i++)
                              -mRcvr(pcli,Ids[i],*th
                                 -e();
    }

    //
    //      h
    //

    BOOL CPostOffic
    {
        bEndThread=
        ResetEvent(h
        for(int i=0; i<
                GetAt(i)-
        DWORD id;
        HANDLE hThread=Creat                                    -c,LPVOID(this),0,&id);
        return hThread!=NULL;
        return AfxBeginThread(Th.                                   :
        return _beginthread((void(                                  -PVOID(this))!=-1;


            -ce::StopTraffic(DWORD dwT
                -=TRUE;
                 -tForSingleObject(hThreadE                                 0;
                     i=0; i<GetSize(); i++)
                      -tAt(i)->OnStop();

    }

    ULONG                                   -ID param)
    {
        CP                                      -)param;
                                                -age 3
```

```
                                        CP
        ->bEndThread) {
          ait for next message o
            *pMsg=ppo->server.Read(
              =NULL) {
                ->Send(*pMsg);
                  e pMsg;

                       i=0; i<ppo->GetSize();

        Se
        retu
}

BOOL CPostOffice::
{
        return server.
```

CPo

H

#inc

#include

class CMsgSt
class CPostOff

//
// Abstract clas

_declspec(dllexport)

id;
ffice& po;

ostOffice;

cerr, cout;

pu

& _po);

WORD
CPostO   po; }

virtual BO   }

virtual void O
virtual void Idl
virtual void OnSto

virtual BOOL Receive(C   0;

lspec(dllexport) CPostOffi

er   server;
eClient> clients;
t;

*cout;

int iLcl=10, int nRem=20,

BOO   RD timeout=1000) ;
e 1

```
                                    CH

                ') { return server.GetN

                    'PWORD Ids, int max);
                      R Name, LPCSTR Compute

                          return SendTo(GetHand
                           Msg& msg);


        B                                  eth);
        BOO                                 h);

        BOOL Sta
        BOOL Stop                                      TE);

        BOOL Close(DW

    e:

        OL bEndThread;
        LE hThreadEnded;
           ULONG __stdcall Th
```

```
                        ORD m, LPCSTR str)
{

        S
        if(s                                    .str,size);
        else
                da
}
    ::CMsg(CMsg& m)
        sndr=m.sndr;
        vr=m.rcvr;
        =m.meth;
        .size;
            emcpy(data=new BYTE[s
                NULL;

                                ize;
};
BOOL CM
{
        MSG
        DWORD
        delete
        if(ReadF                                cbRead,NULL) ||
(GetLastError()==ER
                sndr=
                rcvr=hd
                meth=hdr.
                size=hdr.s.
                if(!size)
                        return
                data=new BYTE[si.
                if(ReadFile(hFile,
        ror()==ERROR_MORE_DATA))
                        return TRUE;
                ALSE;

                    hFile, HANDLE hStop)


                                L;
                                TRUE,FALSE,NULL);
        e.
                                age 1
```

```
ovl;
vents[0];
.hEvent);
hdr,sizeof(hdr),&cbRea
Objects(2,events,FALSE,
r;

l;
t);
ze,&cbRead,&ovl);
k=WaitForMultipleObj                    )==WAIT_OBJECT_0;
    }
    CloseHandle(events[
    eturn bOk;

    e(HANDLE hFile)

                                sizeof(hdr),&cbWritte
    retur.                              &cbWritten,NULL);
}
```

```
cla
{
        WO
        WORD
        WORD                                        bytes)
        PBYTE   da                                    a

blic:

        CMsg(WORD s, WORD                                    ), meth(m), size(0),
        L) {}
        sg(WORD s, WORD r, W
          (CMsg&);
            ~CMsg() { delete []

                const { return sndr;
                nst { return rcvr; }
                t { return meth; }
                  { return size; }
                  return data; }

                    rn PCHAR(data); }
                     PWORD(data); }
                      PDWORD(data); }

        //
        CMsg(                                   dr=rcvr=meth=0; }
        BOOL Re
        BOOL Read
        BOOL Write(

};

dif
```

```
*
CPipe                                    -)
{
}
CPipeClient::~C
{
        if(hPipe!=IN
                Close
ipeClient::Open(LPCS
        try to perform local
        ComputerName[256];
        eng=sizeof(ComputerNa
        mputerName(ComputerNa                    '0';
                zComputer[0]&&lstrcmp                    ?lpszComput
                    "\\pipe\\" + Name;
/*
                                SECURITY_DESCRIPTOR)ne
[SECURITY
        In                          d,SECURITY_DESCRIPTOR_
        SetSe                       E,NULL,FALSE);
        SECURI
        sa.nLengt
        sa.lpSecur
        sa.bInheritHa
*/
        // open a connec
        while(1) {
                hPipe=Create
                        GENERI
                        0,
                        NULL,
                        OPEN_EXISTIN
                        FILE_ATTRIBUTE_
                        NULL);
                if(hPipe!=INVALID_HANDLE_V
                        break;
                    etLastError()!=ERROR_PIPE_
                        break;
                        NamedPipe(Path,20000))
                          eak;
                            HANDLE_VALUE;
}
BOOL CPipe
                                    age 1
```

```
                    ->Write(hPipe);

                    ========================

                                eue()
{
                                    FALSE,NULL);
}

CPipeServer
{
        CloseHa
}

OOL CPipeServer::CMsg
        if(!IsEmpty()) re
        if(Timeout==0) retu
        f(WaitForSingleObjec                                    {
                ResetEvent(hPut
                return TRUE;

                SE;

                        ueue::Get()

                            TRUE);
                              :Get();
}

void CPip                                  p)
{
        CSing
        TIQueue
        SetEvent(
}

BOOL CPipeServer::CMsgQ
{
        CSingleLock sl(&loc
        BOOL bEmpty=TIQueue<C
        return bEmpty;

        ========================                    ============

                ipeServer()
                (10,10)

                eateEvent(NULL,TRUE,FALSE,

}

BOOL CPi                                   ame)
{
                                          e 2
```

```
                    ame;
                      \pipe\\"+Name;
                        thread
                          Stop);
                            n();


CMS                                  imeout)
{
                                        t)?MsgQueue.Get():NULL;
}

BOOL CPipeServ
{
        if(!thread
                retu
        // global stop
        SetEvent(hEvents
        // wait for all th
        DWORD
          MultipleObjects(thre                                    RUE,dwTimeout);
          t==WAIT_FAILED || wt
            for(int i=0; i<thr
                    if(WaitForSi                                    _TIMEOUT) {
                    }
              rn FALSE;

                  hreads.GetSize(); i++
                    e(threads[i]);

}

BOOL CP
{
        DWO
        HANDLE
h=CreateThread(N                                ID(this),CREATE_SUSPEN
        if(h) {
                thre
                Resume
                return
        }
        return FALSE;
        return AfxBeginThread                                   s))!=NULL;
        return _beginthread((vo                                 d,4096,LPVOID(this))!=-1,


        eServer::Connection()


                ty...
                  ESCRIPTOR psd=(PSECURITY_D
                    MIN_LENGTH];
                      ityDescriptor(psd,SECURITY
                        ptorDacl(psd,TRUE,NULL,FAL
                          sa;
                            =psd;

*/
    // C
        HAN
                                                    e 3
```

```
                th,
                   ACCESS_INBOUND | FILE_
                    YPE_MESSAGE |
                     MODE_MESSAGE |

                        D_INSTANCES,



        //
//      BOOL
bConn=ConnectN                                  tLastError()==ERROR_PIPE_
        OVERLAPP
        ovl.hEvent                                  NULL);
        ConnectNamed                                 s FALSE - don't check...
        DWORD dwErr=Ge
        BOOL bConn=FALSE,
        if(dwErr==ERROR_PI
                bConn=TRUE;
          else if(dwErr==ERROR_
                // wait for conn
                HANDLE evnts[2];
                evnts[0]=ovl.hEvent;                                ns connection
                  nts[1]=hEventStop;
                    n=WaitForMultipleObje                                WAIT_OBJECT_0;


                        ext thread
                          n();
                             this connection...
                              CMsg; p->Read(hPipe,h
p=new

    }
        Clos
//      Flush
        Disconn
        CloseHand
        return 0;
}
```

```
H

#in
#includ
#include

class __declsp
{
        HANDLE hPipe
        CString Name;
        CString Path;

          neClient();
            al ~CPipeClient();

                (LPCSTR lpszComputer,
                  CMsg*);

cla                              neServer
{
        C
        CSt

        class __                                ue : public TIQueue<CM
        {
                CMu
                HANDL
        public:
                CMsgQueue
                ~CMsgQueue
                CMsg* Get();
                void Put(CMsg*,
                BOOL IsEmpty();
                BOOL Wait(DWORD Ti
          MsgQueue;

                hEventStop;
                ANDLE> threads;

                        er();

                             zName);
                             ut=INFINITE);
                              =INFINITE);

        L                          ; }

                                  e 1
```

```
                stdcall ConnectionThre.
                    PipeServer*)param)->Con.

};

#endif
```

```
                                        Pos.
                    Defines the initializa.



#ii.
#defi.
#undef
static ch.
#endif

////////////////.                          ''////////////////////////,,
// CPostOfficeApp

EGIN_MESSAGE_MAP(CPo.
        //{{AFX_MSG_MAP(.                                  ' remove mapping macros he.
                // NOTE -                                    blocks of generated code!
                //    DO NO.
        //}}AFX_MSG_MAP
            MAP()

                ''////////////////////,.                              ''////////////
                  construction

                    officeApp()

                        truction code here,
                        icant initialization
}.

////,.                              ''////////////////////,,
// The                                object

CPostOffice.

////////////////.                          ''//////////////////////,.
// CPostOfficeApp

BOOL CPostOfficeApp::.
{
        // Register all c                                nning.  This enables ti.
        //  OLE libraries .                                applications.
        COleObjectFactory::R.

        return TRUE;

                ''////////////////////////,.                          ''////////////
                ry points required for in.

                Object(REFCLSID rclsid, RE

                    TE(AfxGetStaticModuleStat.
                    ClassObject(rclsid, riid,


S.
{
                                    aticModuleState());

}

                                        e 1
```

```
                                    Post
egisterServer, you can
er(void)

        fxGetStaticModuleState(
        dateRegistryAll();

}
```

Pu

Declares the module pa

"CE"

Windows Dynamic Link L

here

```
                                        Postc
                    'n header file for the .


                              'x.h' before including .
#e.
#inclu                              main symbols

//////////,                            '//////////////////////,
// CPostOffic                             'n of this class
// See PostOff.
//

'class CPostOfficeApp

    'ic:
        CPostOfficeApp();

        'es
            lassWizard generated
            'X_VIRTUAL(CPostOffic

            'OL InitInstance();
            'IAL

            'OfficeApp)
                the ClassWizard will .                          'ns here.
                EDIT what you see in                            'de !

};

//////////,                            '//////////////////////,
```

```
                                    Po
        · type library source f
            ocessed by the Make Ty
              ry (PostOffice.tlb).

                    5-0000E81AE78E), versi
]
{


        //                          r APostOffice

        [ uuid(5                        1AE78E) ]
        dispinterfa
        {
                proper
                        /.                          intain property informati

                        //                              ting this section.
                        //{{AF
                        //}}AFX_

                methods:
                        // NOTE - Class                             information here.
                        //    Use extreme                             ction.
                        //{{AFX_ODL_METHOD(
                        id(1)] BSTR GetName(
                          (2)] short GetLocal
                             3)] boolean Start();
                                 boolean Stop(long T
                                 boolean Create(BSTR N
                                    TR GetComputer();
                                       ean Handshake(BSTR Na
Ids, short
                                       MapCErr(short rcvr, s
                                       oCOut(short rcvr, sho
                                          se(long Timeout);


        };

        //  Class informa

        [ uuid(59C57D0C-CE64-
        coclass APostOffice
        {
                [default] dispinter


              APPEND_ODL}}
```

```
                                    Pu
        : type library source f

          rocessed by the Make Ty
            ary (PostOffice.tlb).

                AF-0000E81AE78E), versi
{


        //                              or APostOffice

        [ uuid(                                 81AE78E) ]
        dispinter
        {
                proper
                                                        intain property informati

                //                                      iting this section.
                //{{AF
                //}}AFX_

                methods:
                        // NOTE - Clas                                  information here.
                        //    Use extreme                               ction.
                        //{{AFX_ODL_METHOD
                        [id(1)] BSTR GetName
                          id(2)] short GetLocal
                            3)] boolean Start();
                              ] boolean Stop(long T
                                boolean Create(BSTR N
                                  TR GetComputer();
                                    ean Handshake(BSTR Na
Ids, shor
                                        MapCErr(short rcvr, s
                                        pCOut(short rcvr, sho
                                          se(long Timeout);


        };

        //  Class informa

        [ uuid(59C57D0C-CE64-
        coclass APostOffice
        {
                [default] dispinter


                _APPEND_ODL}}


                                                    e 1
```

```
                                    Pc
                  l C++ generated resourc


                            SYMBOLS
                              //////////////////////,
//
//                                  = 2 resource.
//
#includ

//////////,                                  //////////////////////,
#undef APSTUDI

#ifdef APSTUDIO_IN
//////////////////,                                  //////////////////////////,.
   TEXTINCLUDE

        UDE DISCARDABLE

             h\0"

                  DABLE

                       ""\r\n"


3 TL
BEGIN
    "#d                                  CES\r\n"
    "#def                                 "
    "#define                                \n"
    "#define _                              n"
        "\r\n"
        "#if !def                                  fined(AFX_TARG_ENU)\r\n
        "#ifdef _WI
        "LANGUAGE 9, 1
        "#pragma code_pa
        "#endif\r\n"
        "#include ""res\\Po                                soft Visual C++ edited
ources\r\n"
        "#include ""afxres.rc""                                nts\r\n"
        "#endif"


                  //////////////////////////,.                  ////////
                  UDIO_INVOKED


                            LL) || defined(AFX_TARG_E

LA
#pra
#endif

//////////,                                  //////////////////////////,
                                  ge 1
```

```
                                    Pu


                                NINFO

 FI
#ifde
 FILEFL
#else
 FILEFLAGS
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
EGIN
        BLOCK "StringFi
        BEGIN
        BLOCK "040904B0"
                BEGIN
                        VALUE "
                        VALUE "Fi                                          L\0"
                        VALUE "File
                        VALUE "Interna
                        VALUE "LegalCopy                                     \0"
                        VALUE "LegalTradem
                        VALUE "OriginalFilen
                         ALUE "ProductName",                                 ibrary\0"
                           IE "ProductVersion",



                                        0x409, 1200
    END
END


#endif

#ifndef APSTUDIO_INVOKE
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINC

      e _AFX_NO_SPLITTER_RESOURCE
        AFX_NO_OLE_RESOURCES
        FX_NO_TRACKER_RESOURCES
          NO_PROPERTY_RESOURCES

          RESOURCE_DLL) || defined(

)

          rc2"  // non-Microsoft Vi
          tandard components
#e

//////////////////////////////////////////////////////////////////////////////
#endif
```

```
                                    re
                  -}}
                    -++ generated include f
                        -c


                                  v objects
                  //
                  #ifd
                  #ifnde

                  #define _A                                    129
                  #define _APS_                                 2771
                  #define _APS_NE                                 0
                  #define _APS_NEX
                  #endif
                  #endif
```

urce file that includes
ch will be the pre-comp
l contain the pre-compi

```
                                    St
                  file for standard sys
                    include files that ar
                      quently

#d                                  // Exclude rarely-use

#inclu                                 core and standard compo
#include                               tensions

#ifndef _AFX_
#include <afxo                                 ses
#include <afxodl                                classes
#include <afxdisp.h                             on classes
#endif // _AFX_NO_OLE

      f _AFX_NO_DB_SUPPORT                                    se classes
        <afxdb.h>
          AFX_NO_DB_SUPPORT

          _DAO_SUPPORT                                          s
            h>
              DAO_SUPPORT

                SUPPORT
                                    // M                          mon

                    PORT
```